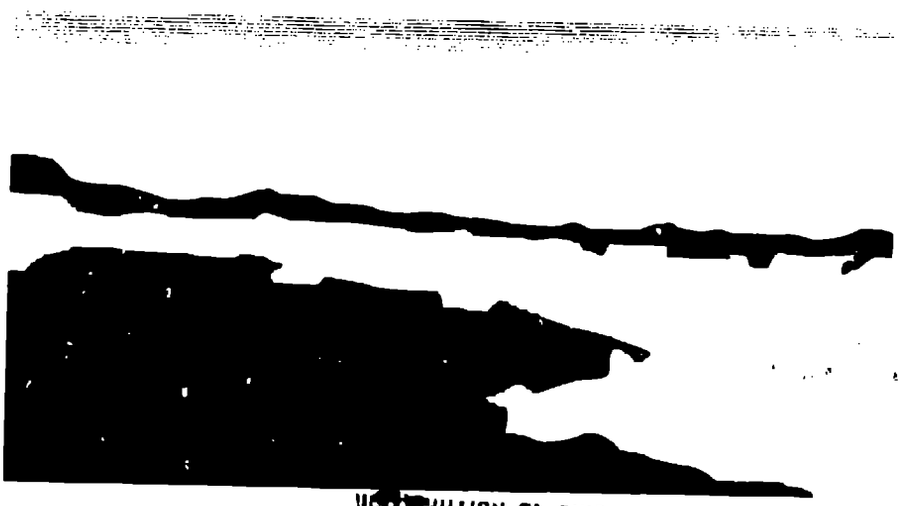
LA-UR- 94-3611

Title: ACCURATE SOLUTION ALGORITHMS FOR INCOMPRESSIBLE MULTIPHASE FLOWS

Author(s): W. J. Rider
D. B. Kothe
S. J. Mosso
J. H. Cerutti

Submitted to: AIAA Aerospace Sciences Meeting
Reno, NV
January 9-12, 1995



Los Alamos
NATIONAL LABORATORY

Accurate Solution Algorithms for Incompressible Multiphase Flows *

William J. Rider, Douglas B. Kothe, S. Jay Mosso, John A. Cerutti,

Los Alamos National Laboratory

Los Alamos, New Mexico, 87545, USA

and

John F. Hochstein

Mechanical Engineering Department

Memphis State University

Memphis, TN, 38152, USA

October 19, 1991

Abstract

A number of advances in modeling multiphase incompressible flow are described. These advances include high-order Godunov projection methods, piecewise linear interface reconstruction and tracking and the continuum surface force model. Examples are given.

I. Introduction

Los Alamos National Laboratory (LANL) has a long history in computational fluid dynamics. A particularly important contribution came in 1965 with the Marker and-Cell (MAC) method for incompressible multiphase flows, which is a method along with its successors that remains a popular choice to this day. An example of a MAC successor is the SOLA-VOF method, which coupled the SOLA (solution algorithm) methodology with a volume tracking method for fluid interfaces. In the nearly thirty years since the inception of these algorithms, numerous advances have been made. We discuss the impact of these advances on the accurate solutions of incompressible multiphase flows.

First, we describe a new algorithm designed to improve the accuracy of the flow solutions. This method is a semi-implicit projection method that uses high-order Godunov methods for advection [1]. The method is second-order accurate in both time and space, and does not suffer from cell Reynolds number-based stability restrictions. The linear systems of equations arising from the projection and diffusion steps are solved via multigrid methods. The algorithm is constructed for a wide range of flow regimes, including multiphase flows where density ratios across interfaces can be arbitrarily large. This feature is greatly facilitated by the use of volume tracking with this method. In conjunction with low error formulations and the filtering of nonsinusoidal modes, the method remains robust for difficult problems. Our method also merges quite naturally with physical models such as the CSF (continuum surface force) model for surface tension [2].

Capturing fluid interfaces with volume tracking techniques remains quite effective, but only when subsequent improvements to the original method (developed 15 years ago) are taken into consideration. An example is the use of piecewise linear or planar approximations to the interface geometry [3]. Volume tracking allows interfaces to be captured and maintained compactly in one cell without imposing restrictions on the topological complexity or the number of interfaces that can be represented. It is also conservative. The ability of the basic volume tracking method to capture interfaces is exhibited with examples and simulation animations. Comparisons are also made where possible with other interface capturing algorithms.

Physical processes specific to and localized at fluid in

*All correspondence should be sent to William J. Rider, Mail Stop B295, Los Alamos National Laboratory, Los Alamos, NM 87545, USA. E-mail: wjr@lanl.gov. This work performed under the auspices of the U.S. Department of Energy by Los Alamos National Laboratory under Contract W-7405-ENG-36. This paper is declared work of the U.S. Government and is not subject to copyright protection in the United States.

terfaces (e.g. surface tension, phase change) are modeled by applying the process to fluid elements everywhere within the interface transition regions. Surface processes are thereby replaced with volume processes whose integral effect properly reproduces the desired interface physics. This methodology is the underlying feature in the CSF method for surface tension, which has proven successful in a variety of studies. The CSF method lifts all topological restrictions (typically inherent in models for surface tension) without sacrificing accuracy, robustness, or reliability. It has been extensively verified and tested in two-dimensional flows through its implementation in a classical algorithm for free surface flows, where complex interface phenomena such as breakup and coalescence have been predicted. This success led to our current focus on extensions and enhancements to the CSF method, namely three-dimensionality, an implicit formulation, and the addition of variable surface tension effects. Computational examples are shown of its capabilities.

II. Projection/Godunov Methods

We solve the equations of incompressible multiphase flow. The equations of motion

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{u}\mathbf{u} + \frac{1}{\rho}\nabla \wp = \frac{1}{\rho}\nabla \cdot \mu\left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right) + \mathbf{F}, \quad (1a)$$

the conservation of mass

$$\nabla \cdot \mathbf{u} = 0, \quad (1b)$$

and density and viscosity transport

$$\frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla \rho = 0, \quad (1c)$$

and

$$\frac{\partial \mu}{\partial t} + \mathbf{u} \cdot \nabla \mu = 0, \quad (1d)$$

In these equations, $\mathbf{u}$ is the velocity vector, $\wp$ is the incompressible pressure, ρ the fluid density, μ the viscosity and $\mathbf{F}$ is the momentum (velocity) source term. $\mathbf{F}$ can include the effects of gravity or surface tension for instance. In our calculations, the transport of density and viscosity is solved with one equation.

$$\frac{\partial f}{\partial t} + \mathbf{u} \cdot \nabla f = 0, \quad (1e)$$

where f is a volume fraction of a fluid having certain properties. This volume fraction is then used to determine the average cell-centered values for density and viscosity (as well as defining the location of a front). Density is recovered from f as follows (for two fluids a and b.

$$\rho(f) = f\rho_a + (1 - f)\rho_b.$$

and similrly viscosity

$$\mu(f) = f\mu_a + (1 - f)\mu_b.$$

These equations will be then solved via a projection method described below.

Our basic goal with projection methods is to advance a velocity field, $\mathbf{V} = (V^x, V^y)^T$ by some convenient means disregarding the solenoidal nature of $\mathbf{V}$, then recover the proper solenoidal velocity field, $\mathbf{V}^d$ ($\nabla \cdot \mathbf{V}^d = 0$). The means to this end is a projection, $\mathcal{P}$, which has the effect

$$\mathbf{V}^d = \mathcal{P}(\mathbf{V}).$$

The projection accomplishes this through the decomposition of the velocity field into parts that are divergence-free and curl-free. This is known as a Hodge or Helmholtz decomposition [4]. The curl-free portion will be denoted by the gradient of a potential, $\nabla\varphi$. This decomposition can be written

$$\mathbf{V} = \mathbf{V}^d + \sigma\nabla\varphi, \quad (2)$$

where $\sigma = 1/\rho$.

Taking the divergence of (2) gives

$$\nabla \cdot \mathbf{V} = \nabla \cdot \mathbf{V}^d + \nabla \cdot \sigma\nabla\varphi \quad \nabla \cdot \mathbf{V} = \nabla \cdot \sigma\nabla\varphi.$$

Once φ has been computed, then the solution can be found through

$$\mathbf{V}^d = \mathbf{V} - \sigma\nabla\varphi. \quad (3)$$

We will be concerned with a class of solution algorithms known as approximate projection methods. These methods are extensions of the classic work of Chorin [5], and its modernization by Bell, Colella and Glaz [6] for solving the incompressible Navier-Stokes equations. These methods are also used to solve equations with varying density [1]. Here we focus on the approximate projection methods introduced by Almgren, Bell and Szymczak [7].

Projection methods can be defined in several ways. In the classical approach the discrete equations behave similarly to the analytic projection operators. Unfortunately, these methods have a number of problems for practical problem. We will be defining discrete methods based on the continuous projections rather than demanding that the discrete system algebraically match the conditions for being a projection. Thus, the most straightforward means to discretize each operator ($\nabla\cdot$, ∇ and $\nabla \quad \sigma\nabla$) will be chosen (not quite true, but nearly).

Two popular approaches to solving the incompressible flow equations are the projection and the pressure Poisson equation (PPE) methods. We will establish the link between these methods and discuss their We have found that the formulation of the Poisson pressure equation can have a profound impact on the solutions quality

The most robust formulation we found is formulated in the following fashion for a fractional step method

$$\nabla \cdot \frac{1}{\rho^{n+\frac{1}{2}}} \nabla \phi = \frac{\nabla \cdot \mathbf{u}^{*\,n+1}}{\Delta t},$$

where ϕ is the pressure and $\mathbf{u}^{*\,n+1}$ is the predicted velocity defined by

$$\mathbf{u}^{*\,n+1} = \left(\mathbf{I} - \frac{\Delta t}{2} \frac{1}{\rho^{n+\frac{1}{2}}} \nabla \cdot \mu^{n+\frac{1}{2}} \left(\nabla + \nabla^T \right) \right)^{-1} \left[\mathbf{u}^n - \Delta t \nabla \cdot (\mathbf{u}\mathbf{u})^{n+\frac{1}{2}} - \frac{\Delta t}{\rho^{n-\frac{1}{2}}} \nabla \phi^{n-\frac{1}{2}} + \Delta t \mathbf{F}^{n+\frac{1}{2}} + \frac{1}{2} \frac{\Delta t}{\rho^{n+\frac{1}{2}}} \nabla \cdot \mu^{n+\frac{1}{2}} \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T \right)^n \right] + \frac{\Delta t}{\rho^{n+\frac{1}{2}}} \nabla \phi^{n-\frac{1}{2}}$$

In terms of a pressure Poisson equation, the form for the right hand side would be

$$\nabla \cdot \left(\frac{1}{\rho^{n+\frac{1}{2}}} \nabla \cdot \mu^{n+\frac{1}{2}} \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T \right)^{n+\frac{1}{2}} - \nabla \cdot (\mathbf{u}\mathbf{u})^{n+\frac{1}{2}} + \mathbf{F}^{n+\frac{1}{2}} + \frac{\mathbf{u}^n}{\Delta t} \right)$$

This option controls the growth of the discrete divergence without sacrificing the accuracy of the method. A full exposition on this subject is given in [8, 9].

When using approximate projections it is important to control the growth of divergence errors. This can be accomplished through the use of filtering. A number of filters have been devised to remove or damp spurious non-divergence-free modes from the solution. Unobstructed these modes can seriously damage the accuracy and stability of calculations. This is especially true in several cases: large density jumps, local source terms (such as surface tension) and for long-term integrations. These filters are described more fully in [8, 10].

The use of nonoperator split high-order Godunov advection is also important to the method's success. This method avoids the use of operator splitting and allows the consistent use of high-order monotone advection in incompressible flows. The method also has excellent phase error properties and compares favorably with higher order Runge-Kutta methods on typical grids.

Space does not permit an adequate description of the method and good descriptions appear in several sources [11, 6, 1, 12, 7, 8]. The basic point of the method is to implement a second-order Taylor series in both time and space expansion for the dependent variables in multidimensional form. Variables are constructed to cell-edges (where they will be multivalued) and then the upwind values are propagated in the solution. During this procedure the complete effect of the governing equations is used to approximate the time derivatives (source terms, pressure gradients and the incompressibility constraint). The use of the incompressibility constaint leads to the "MAC" projection where time-centered normal velocities are used to compute a potential field that corrects them to be divergence-free. This allows the method to use a divergence-free velocity field for the purpose of advection (even with an approximate projection) and maintain second-order accuracy.

Several points are important to emphasize. This method is not bound by a stability restriction related to the Reynolds number. Thus this method can be used to compute both viscous and inviscid flows. Results show that the method can resolve flows up to a grid Reynolds number of 10. Beyond this the solver will degrade in a graceful fashion and produce physical, but not necessarily accurate results (in the classical sense). The advection method is also robust in cases where the flow contains discontinuities (such as a shear).

III. Interface Kinematics

All of the methods in this paper solve the equation

$$\frac{\partial f}{\partial t} + \mathbf{u} \cdot \nabla f = 0, \tag{4a}$$

where f is some scalar carrying interface or "color" information. An equivalent equation for incompressible flows is

$$\frac{\partial f}{\partial t} + \nabla \cdot (\mathbf{u} f) = 0, \tag{4b}$$

since $\nabla \cdot \mathbf{u} = 0$.

A particle method is perhaps the most straightforward, drawing upon recent advances in PIC algorithms [13]. Particles are assigned a mass according to the density of the fluid in which they reside and a volume (hence size) according to the interpolation function chosen to interpolate quantities to and from the computational grid. While this method provides a superior multidimensional grid-independent advection scheme, there are as a result some practical difficulties, namely the cost and accuracy associated with interpolating the particle information to an Eulerian grid [14, 15].

Simply discretising (4) with a high resolution finite difference scheme is quite appealing. An advection algorithm is typically an integral part of a flow solver. This is done in the methods presented in [1, 16]. From the advances in high speed flows in the last decade, there are a number of methods that minimize numerical dissipation. An example of this are high-order Godunov methods in particular PPM [17, 18].

Problems with the numerical dissipation (leading to a thickening interface) led researchers to propose an ingenious compromise. The level set methods could be implemented with the same difference techniques already well developed for advection, but without allowing the interface to smear. The interface is defined as the zero

level set of a distance function, c, from that interface. Instead of (4), the following equation is solved

$$\frac{\partial c}{\partial t} + \mathbf{u} \cdot \nabla c = 0.$$

We also study improvements suggested by Sussman, Smereka and Osher [19].

VOF methods have been used for several decades starting at the national laboratories (Livermore [20] and Los Alamos [21] and later Sandia [22]). The earlier work is typified by the SLIC [23] algorithm and the original method with the *monker* VOF [24]. In each of these methods the interface is designated as a straight line in a cell defined by the volume of a given fluid in that cell. Youngs [3] improved the general method greatly by allowing the reconstruction of the interface to be multidimensional and linear in nature. Youngs further extended his method to three dimension in [25]. Recently, Pilliod and Puckett have improved the accuracy [26, 27, 28]. Here we refer to this method as the piecewise linear interface calculation (PLIC).

Our results indicate that the PLIC method produces the best overall results. Capturing methods although simple are expensive when implemented to achieve the best resolution of interfaces and still do not match the resolution of PLIC. Level sets methods although conceptually appealing suffer from a number of detriments. First among these is the lack of volume (mass) conservation. Cost is also an issue. Particle methods offer excellent, numerical diffusion-free results, but have two problems: cost and the problem of mapping the particles to a grid in a manner free of oscillatory modes.

These results become quite obvious when nontrivial pure advection problems are used to evaluate the methods. By nontrivial we mean that the flow field varys in space and perhaps time and contains vorticity. For the purposes of solving multiphase flow we favor method that preserve symmetries in the solution naturally. For this reason we have developed an unsplit (not operator split) version of the PLIC method that will be used below for some example calculations.

IV. Interface Dynamics

Interface dynamics such as surface tension are modeled with a localized volume force prescribed by the CSF model.[2] Ideally suited for interfaces of arbitrary and changing topology, the CSF model's central theme of volume reformulation is a new and radical departure from previous traditional finite difference representations of interfacial phenomena such as surface tension. The basic premise of the CSF model is to replace interfacial surface phenomena, normally applied via a discrete boundary condition, as smoothly varying volumetric forces derived from a product of the appropriate interfacial physics per unit area and a numerical approximation to the surface (interface) delta function. The resulting simplicity, accuracy, and robustness, often a rare yet sought after combination for numerical models, has led to its widespread and popular use in modeling complex interfacial flows that were in many cases previously intractable. For modeling surface tension, the relevant surface physics is a force per unit area arising from local interface curvature and local (tangential) variations in the surface tension coefficient. Surface phenomena other than surface tension can also be encapsulated easily within the CSF model, examples are phase change and momentum exchange[], where the surface physics are mass and momentum flux, respectively, transferred across the interface.

In this section, we briefly review the theory of the CSF model, then discuss some important issues pertaining to the accurate formulation of the necessary discrete operators. We also outline the CSF enhancements and improvements that are currently being studied which should allow a more efficient modeling of a wider variety of interfacial flows.

In the CSF model, surface tension is reformulated as a volume force $\mathbf{F}_s$ satisfying

$$\lim_{h \to 0} \int_{\Delta V} \mathbf{F}_s(\vec{x}) d^3x = \int_{\Delta S} \mathbf{f}_s(\vec{x}_s) dS, \qquad (5)$$

where $\vec{x}_s$ is a point on the surface, $\mathbf{f}_s(\vec{x})$ the surface tension force per unit interfacial area,

$$\mathbf{f}_s(\vec{x}_s) = \sigma \kappa(\vec{x}_s) n(\vec{x}_s), \qquad (6)$$

and h is a length comparable to the resolution afforded by a computational mesh with spacing Δx. The area integral is over the portion ΔS of the surface lying within the small volume of integration ΔV.

Since interfaces having surface tension are tracked with the aforementioned PLIC volume tracking method, their topology will not in general align with logical mesh coordinates. Discontinuous interfaces are therefore represented in the computational domain as finite thickness transition regions within which fluid volume fractions vary smoothly from zero to one over a distance of $\mathcal{O}(h)$. The volume force, nonzero only within these transition regions, is given in the CSF model by[2]

$$\mathbf{F}_s(\vec{x}) = \sigma \kappa(\vec{x}) \frac{\nabla \tilde{c}(\vec{x})}{[c]}, \qquad (7)$$

where σ is the surface tension coefficient, $\tilde{f}$ is VOF function, and κ the mean interface curvature, given by:[7]

$$\kappa = -(\nabla \cdot n), \qquad (8)$$

where the unit normal n,

$$n = \frac{\hat{n}}{|\hat{n}|}. \qquad (9)$$

is derived from a normal vector $\vec{n}$.

$$\vec{n} = \nabla f . \tag{10}$$

that is the gradient of VOF data. Because the curvature is proportional to the second derivatives of the VOF function, surface force modeling can be very sensitive to small fluctuations in f, possibly amplifying them

The CSF formulation makes use of the fact that numerical models of discontinuities in finite volume and finite difference schemes are really continuous transitions within which the fluid properties, vary smoothly from one fluid to another over a distance of $\mathcal{O}(h)$, where h is a length comparable to the resolution afforded by a computational mesh with spacing δx. It is not appropriate, therefore, to apply in finite difference schemes a pressure jump induced by surface tension at a free surface "discontinuity." Surface tension should act everywhere within the transition region, namely through the volume force $\mathbf{F}_s$.

The volume force in the CSF model is easily calculated by taking first and second order spatial derivatives of the fluid volume fractions. At each point within the free surface transition region, a cell-centered value $\mathbf{F}_s$ is defined which is proportional to the curvature κ of the constant VOF surface at that point. The force is normalized to recover the conventional description of surface tension as the local product $\kappa h = 0$. Its line integral directed normally through the free surface transition region is approximately equal to the surface pressure in equation Wall adhesion is incorporated by enforcing a simple boundary condition.

Surface tension modeled with the continuum method eliminates the need for interface reconstruction, so restrictions on the number, complexity, or dynamic evolution of interfaces having surface tension are not imposed. Direct comparisons between modeling surface tension with the CSF model and with a popular interface reconstruction model show that the CSF model makes more accurate use of volume fraction data.[2] The volume force always tends to force the free surface to seek a minimum surface energy configuration. Reconstruction models, on the other hand, tend to induce numerical noise from computed graininess in the surface pressures, often leading to unphysical free surface disruptions. In addition to providing a more accurate finite difference representation of surface tension without the topological restrictions, the CSF model is easy to implement computationally. Surface tension is easily included by calculating and applying an extra body force, $\mathbf{F}_s$, in the momentum equation. A small fraction of the total CPU time (few percent) is spent computing surface tension effects.

The force $\mathbf{F}_s$ resides at cell centers in this scheme, collocated with all other fluid quantities. In practice we find that the use of more free interface information information (a larger stencil) leads to a better estimate of curvature. This appears to be the case for $\mathbf{F}_s$ at cell centers, which requires a 6 cell stencil when both components of interface normal $\vec{n}$ are collocated at faces. A cell-centered force is then obtained by summing over cell faces, which brings the effective stencil to 9 cells.

The "optimal" discrete approximation of $\mathbf{F}_s$ in the CSF model is likely to be one that properly accounts for the duality of needing a large stencil for accurate curvature estimates and a compact stencil for maintaining the locality of volume force. Our current operators for $\mathbf{F}_s$ are found to give the god results in practice. They are not necessarily optimal, and certainly by no means unique. Some comparisons with theory using various difference expressions for $\mathbf{F}_s$ can be found elsewhere.[2]

A smoothed value of the curvature κ can be computed from equation ?? by using a smoothed VOF function $\tilde{f}$ to derive the normals needed in equation 10. The smoothed VOF function is computed with an expression of the form

$$\tilde{f}_{i,j} = \sum_{i',j'=1}^{n} f_{i',j'} \, \mathcal{S}^{(\ell)}(x'_{i'} - x_i; h) \mathcal{S}^{(\ell)}(y'_{j'} - y_j; h) , \tag{11}$$

where $\mathcal{S}^{(\ell)}$ is a B-spline[?] of degree ℓ, having finite support, $\mathcal{S}^{(\ell)}(x' - x; h) \neq 0$ only for $x' - x < (\ell + 1)h/2$. This smoothing tends to mitigate the high wavenumber contributions to κ, which may or may not be a result of discretization errors. It should, therefore, be used with caution because the real free surface geometry might be unphysically mollified. Examples of the smoothing effects can be found in the CSF paper.[2]

The representation of surface tension in the CSF model as a time t^n body force is linearly stable only for time steps smaller than a certain maximum allowable value δt_s necessary to resolve the propagation of capillary waves.[2] evaluated as

$$\delta t < \delta t_s = \left[\frac{\rho J_{\min}^{3/2}}{4\pi\sigma} \right]^{1/2} . \tag{12}$$

where $J_{\min}$ is the minimum mesh cell Jacobian.

Wall adhesion is the surface force acting on fluid interfaces at points of contact with "walls," which are static, rigid boundaries. Wall adhesion forces are calculated in the same manner as volume forces due to surface tension are calculated, except that a boundary condition is applied to the free surface unit normal n prior to evaluating equation 10. The condition is applied only to those normals lying on a rigid boundary. Those forces $\mathbf{F}_s$ attributed to wall adhesion are therefore only in cells along a boundary.

The wall adhesion boundary condition becomes an expression for the unit free surface normal n at points of contact $\vec{x}_w$ along the wall:

$$n = n_w \cos\theta_{eq} + t_w \sin\theta_{eq} , \tag{13}$$

where θ_{sl} is the static contact angle between the fluid and the wall, n_w is the unit wall normal directed into the wall, and t_w is tangent to the wall, normal to the contact line between the free surface and the wall at $\vec{x}_w$. The equation uses the geometric identity that θ_{sl}, defined as the angle between the tangent to the fluid and t_w, is also the angle between n_w and n. The unit tangent t_w is directed into the fluid, and is computed from equation 10 with the VOF function f reflected at the wall. The angle θ_{sl} is not a fluid material property, but a system property, depending also on properties of the wall itself. The value of θ_{sl} is measured experimentally when the fluid is at rest. We emphasize that the wall adhesion boundary condition above is applied regardless of whether or not the angle the fluid interface with the wall is actually equal to θ_{sl}.

V. Algorithm Summary

Algorithm 1 *(Godunov/Projection Method*

1. Start with data $\mathbf{u}^n$, f^n, $\sigma^{n+\frac{1}{2}}$.
2. Compute the time- and edge-centered velocity field via the Godunov procedure based soley on data at time n.
3. Solve the "MAC" projection and correct the time- and edge-centered velocity field so that $D\mathbf{u}^{n+\frac{1}{2}} = 0$.
4. Using $\mathbf{u}^{n+\frac{1}{2}}$ update the volume fraction data from n to $n+1$.
5. Using f^{n+1} compute ρ^{n+1} and μ^{n+1}
6. Compute the CSF source term using $f^{n+\frac{1}{2}} = (f^n + f^{n+1})/2$.
7. Update $\mathbf{u}^n$ to $\mathbf{u}^{*,n+1}$ with advection, diffusion and CSF sources.
8. Solve the projection $L_\rho \sigma^{n+\frac{1}{2}} = D\mathbf{u}^{*,n+1}$ and correct the velocities to $\mathbf{u}^{n+1}$.
9. Filter the advanced time velocities.
10. Begin the next cycle

Several comments are in order regarding the method we use. The stability restriction on the method is the sum of the usual CFL restriction and the restriction based on the capillary wave speed. Linear algebra is computed using a multigrid method. One for the single pressure equations and a second method for the coupled velocity diffusion equations. Our advection method is based on a variant of the PPM method (uses PPM as the one-dimensional building block for the unsplit scheme). Volume tracking is computed using an unsplit methods using a modified version of Pilliod and Puckett's fast least squares algorithm.

VI. Numerical Examples

Here we will present three numerical examples fo the methods we have described. The first example is a drop falling into a pool and the subsequent splash. Our second problem is a Rayleigh-Taylor instability. Finally, we show a Kelvin Helmholtz instability as an idealized flow of wind over water.

The drop and splash is set in a 1×1 box with the drop and pool density being 800 times higher than the rest of the medium. The drop has a radius of 0.15 and is set at (0.50, 0.75) Each boundary is a solid wall. The computational grid is 128×128, and is at rest initially. The pool covers the bottom of the domain and has a depth of 0.25 units. The heavy fluid has a viscosity that is 832.5 times higher than the surrounding medium. The high density fluid viscosity is 0.001 and the Bond number, $\rho g l^2/\sigma$, is 1000. Gravity is set to -1. The wall adhesion model is used with and equilibrium angle of 15 degrees.

The Rayleigh-Taylor instability simulates air above helium giving a density ratio of 7.25:1. The air's viscosity is 1.06 times larger than the helium. The computational grid is 64×256. In the horizontal direction the domain is periodic. Gravity is set to -1. A perturbation is applied to the vertical velocity of

$$v = 0.005\left[\cos(2\pi x) + 1\right].$$

The calculation uses no surface tension.

The third problem is a Kelvin-Helmholtz instability. The physical properties are the same as the first problem. The lower part of the domain is filled with water with air above. The computational grid is 256×64. In the horizontal direction the domain is periodic. Gravity is set to -1. A perturbation is applied to the vertical velocity of

$$v = 0.005\left[\cos(2\pi x) + 1\right].$$

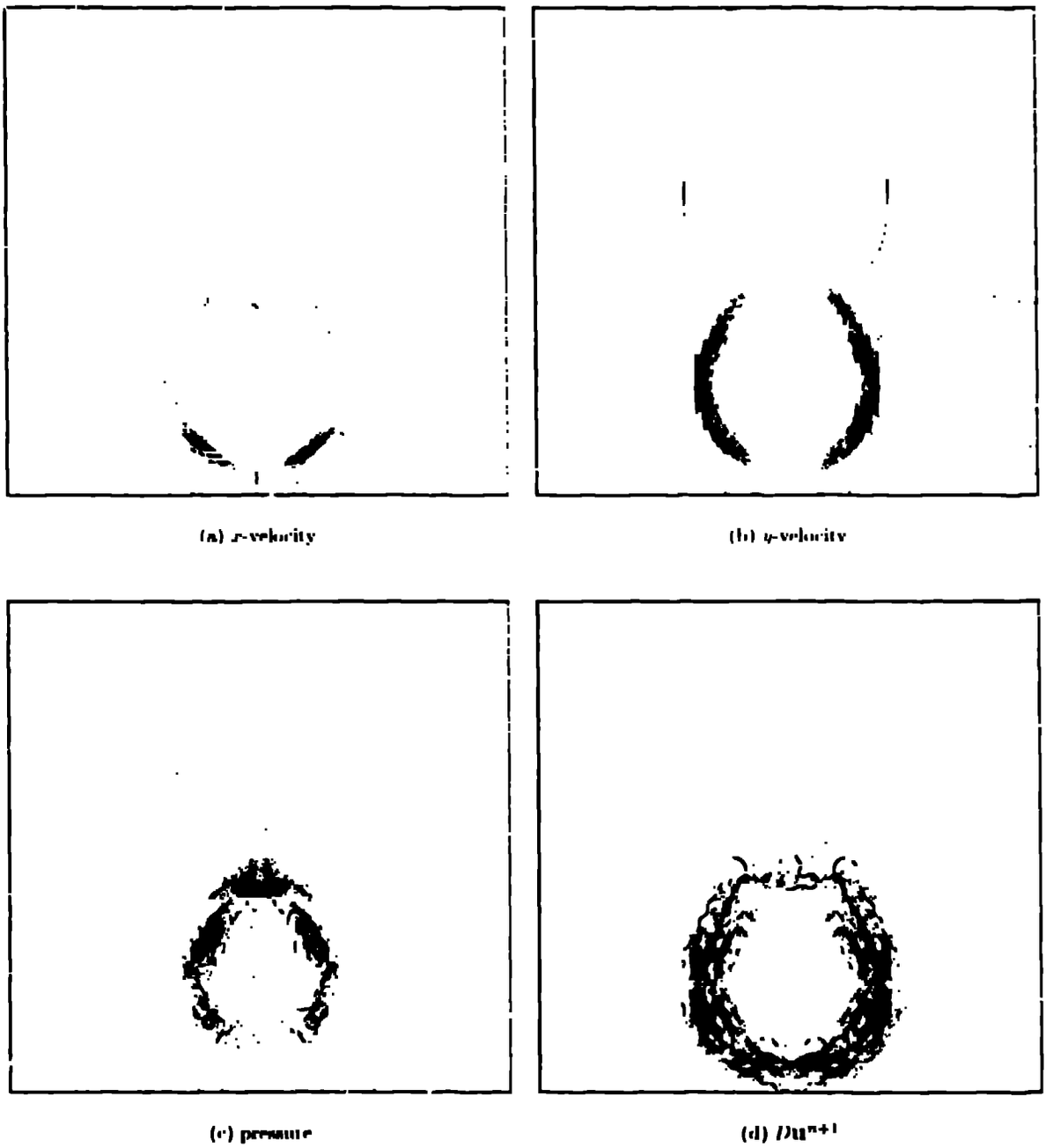
Surface tension is applied to the problem

VII. Future Directions

References

[1] J. B. Bell and D. L. Marcus. A second-order projection method variable-density flows. *Journal of Computational Physics*, 101:334 348, 1992.

[2] J. U. Brackbill, D. B. Kothe, and C. Zemach. A continuum method for modeling surface tension



(a) x-velocity

(b) y-velocity

(c) pressure

(d) Du^{n+1}

Figure 1: Inviscid discontinuous drop results with vertex and velocity filters using a pressure projection of $Du^n/\Delta t$ and the nine-point formulation.

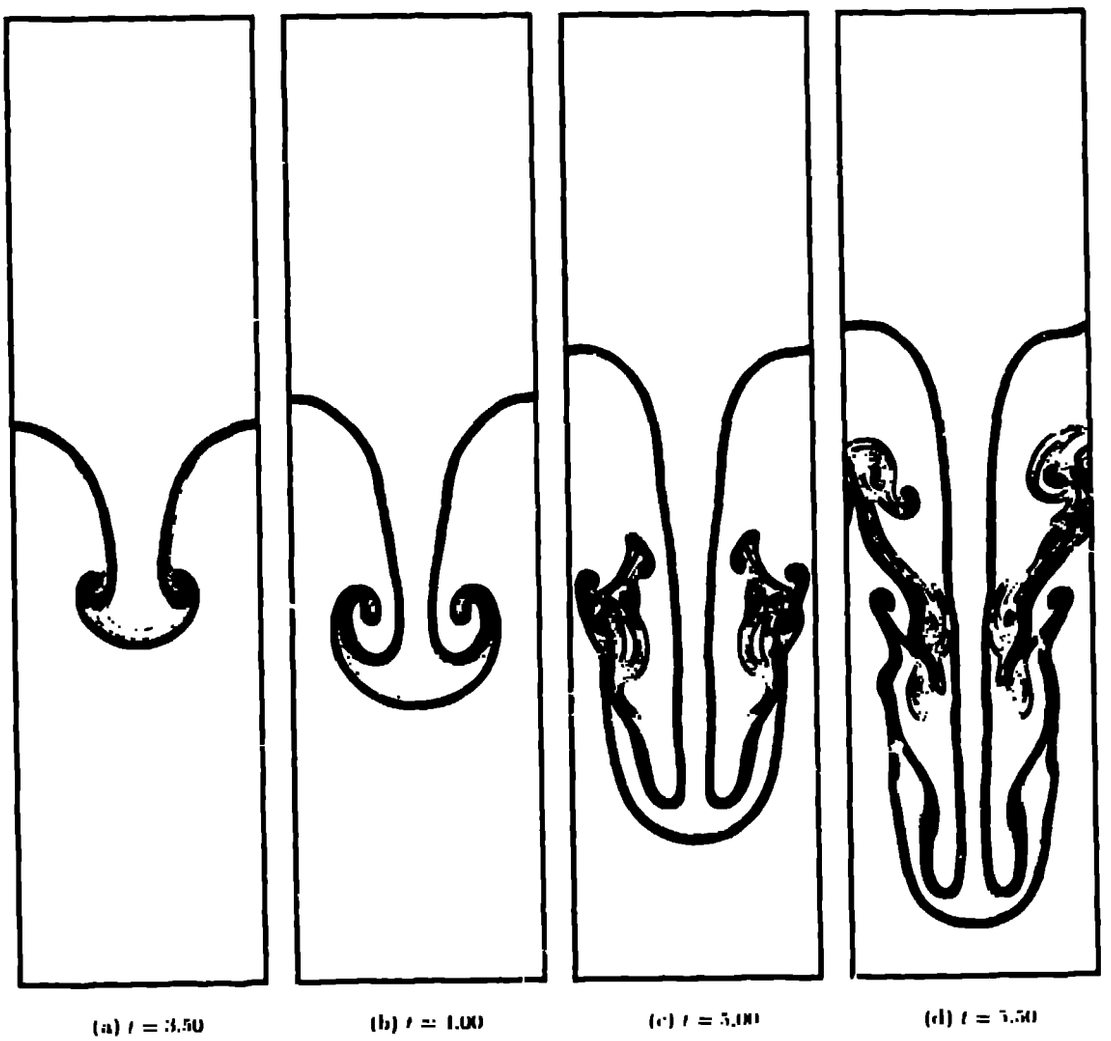


Figure 2: Rayleigh-Taylor instability evolution. The density profile is shown at each time.

(a) $t = 7.15$

(b) $t = 7.36$

(c) $t = 7.15$

(d) $t = 7.36$

Figure 3: Variable density ($\beta = 7$) shear-layer with inflow from the left and outflow at the right with a 512×128 grid. The density profile is shown.

Journal of Computational Physics, 100:335–354, 1992.

[3] D. L. Youngs. Time-dependent multi-material flow with large fluid distortion. In K. W. Morton and M. J. Baines, editors, *Numerical Methods for Fluid Dynamics*, pages 273–285, 1982.

[4] A. J. Chorin and J. Marsden. *A Mathematical Introduction to Fluid Mechanics*. Springer-Verlag, 1993.

[5] A. J. Chorin. Numerical solution of the Navier-Stokes equations. *Mathematics of Computation*, 22:745–762, 1968.

[6] J. B. Bell, P. Colella, and H. M. Glaz. A second-order projection method for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 85:257–283, 1989.

[7] A. S. Almgren, J. B. Bell, and W. G. Szymczak. A numerical method for the incompressible Navier-Stokes equations based on an approximate projection. Technical Report UCRL-JC-112842, Lawrence Livermore National Laboratory, 1993.

[8] W. J. Rider. Approximate projection methods for incompressible flow: Implementation, variants and robustness. Technical Report LA-UR-2000, Los Alamos National Laboratory, 1994. Available on World Wide Web at `http://www.c3.lanl.gov-/cic3/publications/main.shtml`.

[9] W. J. Rider. The robust formulation of approximate projection methods for incompressible flows. Technical Report LA-UR-3015, Los Alamos National Laboratory, 1994. Available on World Wide Web at `http://www.c3.lanl.gov/cic3-/publications/main.shtml`.

[10] W. J. Rider. Filtering nonsolenoidal modes in numerical solutions of incompressible flows. Technical Report LA-UR-3011, Los Alamos National Laboratory, 1994. Available on World Wide Web at `http://www.c3.lanl.gov/cic3-/publications/main.shtml`.

[11] P. Colella. Multidimensional upwind methods for hyperbolic conservation laws. *Journal of Computational Physics*, 87:171–200, 1990.

[12] J. B. Bell, P. Colella, and L. Howell. An efficient second-order projection method for viscous incompressible flow. In D. Kwak, editor, *Proceedings of the AIAA Tenth Computational Fluid Dynamics Conference*, pages 360–367, 1991. AIAA Paper 91-1560.

[13] J. U. Brackbill, D. B. Kothe, and H. M. Ruppel. FLIP: A low-dissipation, particle-in-cell method for fluid flow. *Computer Physics Communications*, 48:25–38, 1988.

[14] S. O. Unverdi and G. Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics*, 100:25–37, 1992.

[15] S. O. Unverdi and G. Tryggvason. Computations of multi-fluid flows. *Physica D*, 60:70–83, 1992.

[16] W. G. Szymczak, J. C. W. Rogers, J. M. Solomon, and A. E. Berger. A numerical algorithm for hydrodynamic free boundary problems. *Journal of Computational Physics*, 106:319–336, 1993.

[17] P. Colella and P. Woodward. The piecewise parabolic method (PPM) for gas-dynamical simulations. *Journal of Computational Physics*, 54:174–201, 1984.

[18] P. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *Journal of Computational Physics*, 54:115–173, 1984.

[19] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114:146–159, 1994.

[20] R. DeBar. Fundamentals of the KRAKEN code. Technical Report UCIR-760, LLNL, 1974.

[21] K. S. Holian, S. J. Mosso, D. A. Mandell, and R. Henninger. MESA: A 3-D computer code for armor/anti-armor applications. Technical Report LA-UR-91-569, Los Alamos National Laboratory, 1991.

[22] J. S. Perry, K. G. Budge, M. K. W. Wong, and T. G. Trucano. Rhale: A 3-D MMALE code for unstructured grids. In ASME, editor, *Advanced Computational Methods for Material Modeling. AMD-Vol. 180/PVP-Vol. 268*, pages 159–174, 1993.

[23] W. F. Noh and P. R. Woodward. SLIC (simple line interface method). In A. I. van de Vooren and P. J. Zandbergen, editors, *Lecture Notes in Physics 59*, pages 330–340, 1976.

[24] C. W. Hirt and B. D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39:201–225, 1981.

[25] D. L. Youngs. An interface tracking method for a 3D Eulerian hydrodynamics code. Technical Report 44/92/35, AWRE, 1984.

[26] J. J. Pilliod, Jr. and E. G. Puckett. Second-Order Volume-of-Fluid Algorithms for Tracking Material Interfaces. In preparation.

[27] E. G. Puckett. A volume of fluid interface tracking algorithm with applications to computing shock wave refraction. In *Proceedings of the Fourth International Symposium on Computational Fluid Dynamics*, pages 933–938, 1991.

[28] E. G. Puckett and J. S. Saltzman. A 3D adaptive mesh refinement algorithm for multimaterial gas dynamics. *Physica D*, 60:84–93, 1992.